CLAIMS

We claim:

1. A method for control and management of communication traffic, comprising the steps of:

expressing access rules as filters referencing system kernel data;

for outbound processing, determining source application indicia;

for inbound packet processing, executing a look-ahead function to determine target application indicia; and

responsive to said source or target application indicia, executing filter processing.

2. The method of claim 1, further comprising the steps of executing said determining and executing steps within a

kernel filtering function upon encountering a filter selector field referencing kernel data not included in said packet.

3. The method of claim 1, said filter processing including the steps of:

determining a task or thread identifier;

based on said task or thread identifier, determining a process or job identifier; and

based on said process or job identifier, determining job or process attributes for filter processing.

4. The method of claim 1, said filter processing including the steps of:

determining a user identifier; and

based on said user identifier, determining user attributes for filter processing.

5. The method of claim 3, further comprising the step of determining from said task identifier a work control block containing said process or job identifier.

6. The method of claim 1, further comprising the steps for inbound processing of:

passing an inbound packet to a sockets layer to identify said target application.

7. The method of claim 6, further comprising the step of marking said inbound packet as not deliverable before passing it to said sockets layer.

8. The method of claim 1, further comprising the steps of:

delivering to said filters infrastructure access rules for defining security context.

9. The method of claim 8, said infrastructure including

logging, auditing, and filter rule load controls.

10. A method for control and management of aspects of communication traffic within filtering, comprising the steps of:

receiving IP packet data into a TCP/IP protocol stack executing within a system kernel

executing filtering code within said system kernel with respect to non-IP packet data accessed within said system kernel outside of said TCP/IP protocol stack.

11. The method of claim 10, said non-IP packet data including context data regarding said IP packet.

12. The method of claim 10, said non-IP packet data including data specific to a task generating said non-IP packet data.

13. The method of claim 10, said non-IP packet data

including data specific to a task that will receive said IP packet.

14. The method of claim 11, said context data including packet arrival interface indicia.

15. The method of claim 11, said context data including packet arrival time-of-day indicia.

16. The method of claim 10, further comprising the steps of:

establishing a tunnel between two IP address limiting traffic to applications bound to ports at each end of said tunnel;

said filtering code accessing filtering attributes further limiting traffic selectively to job indicia.

17. The method of claim 10, further comprising the steps of:

establishing a tunnel between two IP address limiting traffic to applications bound to ports at each end of said tunnel; and

said filtering code accessing filtering attributes further limiting traffic selectively to user identification indicia.

18. A method for centralizing system-wide communication management and control within filter rules, comprising the steps of:

providing filter statements syntax for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values; and

said selector referencing data that does not exist in IP packets.

19. The method of claim 18, said parameters selectively including userid, user profile, user class, user group, user group authority, user special authority, job name, process

name, job group, job class, job priority, other job or process attributes, and date & time.

20. The method of claim 18, said filters statements being provided within a user interface to said system.

21. The method of claim 18, further comprising the steps of:

establishing a tunnel between two IP address limiting traffic to applications bound to ports at each end of said tunnel;

said filtering code accessing filtering attributes further limiting traffic selectively to job indicia; and

operating said filtering code within a kernel filtering function upon encountering a filter selector field referencing kernel data not included in said traffic.

22. A method for traversing a portion only of a protocol

stack to disallow selective IP packet traffic, comprising the steps of:

receiving a packet in the kernel of the operating system of a first node from an application, said kernel including a filter processor;

for inbound packet processing to a first node from a second node, executing a look-ahead function in the system kernel of said first node to determining a target application;

for both said inbound packet processing, and for outbound packet processing from said first node to said second node, executing within said kernel the steps of

processing said packet by determining a task ID;

responsive to said task ID, determining a corresponding work control block;

determining a user ID, process or job identifier from said work control block;

from the user ID, process or job identifier

selectively determining attributes for said user process or job; and

passing said attributes to said filter processor for managing and controlling communication traffic.

23. A method for expressing access rules as filters, comprising the steps of:

providing a filter statements syntax for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values; and

said selector referencing data that does not exist in IP packets for controlling access to an application.

24. A method for managing and controlling communication traffic by centralizing access rules in filters executing within and referencing data available in system kernels, comprising the steps for outbound packet processing from a first node to a second node of:

receiving said packet in the kernel of the operating system of said first node from an application or process at said first node;

processing said packet by determining a task ID;

responsive to said task ID, determining a corresponding work control block;

responsive to said work control block, determining a process or job identifier;

responsive to said process or job identifier, determining job or process attributes.

25. The method of claim 24, further comprising the steps for inbound packet processing from said second node to said first node of:

initially operating said kernel at said first node to determine a target application for said packet at said first node.

26. The method of claim 25, said initially operating step comprising executing a look-ahead function.

27. The method of claim 26, said look-ahead function including the steps of operating a filter function to request of a sockets layer the identity of an application to which said sockets layer would pass said packet.

28. The method of claim 27, further comprising the step of marking said packet as non-deliverable and thereafter passing said packet to said sockets layer to identify said application.

29. A method for managing and controlling communication traffic by centralizing the access rules, comprising the steps for outbound packet processing from a first node to a second node of:

receiving said packet in the kernel of the operating system of said first node from an application or process at said first node, said kernel including a filter processor;

processing said packet by determining a task ID;

responsive to said task ID, determining a corresponding work control block;

determining a user ID control block from said work control block;

from the user ID control block determining attributes for said user; and

passing said attributes to said filter processor for managing and controlling communication traffic.

30. The method of claim 29, further comprising the steps for inbound packet processing from said second node to said first node of:

initially operating said kernel at said first node to determine a target application for said packet at said first node.

31. The method of claim 30, said initially operating step

comprising executing a look-ahead function.

32. The method of claim 31, said look-ahead function including the steps of operating a filter function to request of a sockets layer the identity of an application to which said sockets layer would pass said packet.

33. The method of claim 32, further comprising the step of marking said packet as non-deliverable and thereafter passing said packet to said sockets layer to identify said application.

34. A method for control and management of communication traffic with respect to a system node, comprising the steps of:

receiving at said system node an inbound packet; and

executing within a protocol stack of the system kernel of said system node a filtering function identifying for said inbound packet a filter referencing non-packet data; and

responsive to said filter, executing a look-ahead function for identifying a target application for said inbound packet.

35. The look-ahead function of the method of claim 34 further comprising the steps of:

passing to a transport layer function identified by an IP header a packet marked non-deliverable for determining which user-level process or job is to receive said packet;

receiving from said transport layer an application layer task identifier for said user-level process or job; and thereafter

passing said packet marked by said task identifier to said transport layer for delivery to said application layer task.

36. System for control and management of communication traffic, comprising:

a system kernel including a filter function and stack data;

said filter function including a filter selectively referencing said stack data for expressing access rules;

said filter function being responsive to receipt of an outbound packet for determining a source application;

said filter function being responsive to receipt of an inbound packet processing for executing a look-ahead function to determine a target application; and

said filter function being responsive to said source or target application for executing filter processing.

37. A system for control and management of aspects of communication traffic within filtering, comprising:

a system kernel;

a protocol stack executing within said system kernel for receiving IP packet data; and

filtering code within said system kernel operable with respect to non-IP packet data accessed within said system kernel outside of said protocol stack for controlling and managing said aspects of communication traffic.

38. A system for centralizing system-wide communication management and control within filter rules, comprising:

filter statements having a syntax for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values; and

said selector referencing data that does not exist in IP packets.

39. A system for traversing a portion only of a protocol stack to disallow selective IP packet traffic, comprising:

a system kernel;

a filter processor executing within said system kernel;

said filter processor responsive to an inbound packet for executing a look-ahead function for determining a target application;

said filter processor responsive to both inbound and outbound packets for

processing said packet by determining a task ID;

responsive to said task ID, determining a corresponding work control block;

determining a user ID, process or job identifier from said work control block;

from the user ID, process or job identifier selectively determining attributes for said user process or job; and

passing said attributes to said filter processor for managing and controlling communication traffic.

40. A system for expressing access rules as filters,

comprising:

a filter statements for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values; and

said selector referencing data that does not exist in IP packets for controlling access to an application.

41. A system for managing and controlling communication traffic by centralizing access rules in filters executing within and referencing data available in system kernels, comprising:

code for receiving a packet in the kernel of the operating system of a first node from an application or process at said first node;

code for processing said packet by determining a task ID;

code responsive to said task ID for determining a corresponding work control block;

code responsive to said work control block for determining a process or job identifier; and

code responsive to said process or job identifier for determining job or process attributes.

42. A system for managing and controlling communication traffic by centralizing access rules, comprising:

a first system node;

a second system node;

a kernel of the operating system of said first system node including a kernel filter processor;

said kernel for receiving from an application or process at said first system node a packet for communication to said second system node;

said kernel further for processing said packet by determining a task ID; responsive to said task ID, determining a corresponding work control block; determining a user ID control block from said work

control block; from the user ID control block determining attributes for said user; and passing said attributes to said system kernel filter processor for managing and controlling communication traffic.

43. A system for control and management of communication traffic with respect to a system node, comprising:

a filtering function executing within a protocol stack of the system kernel of said system node identifying for an inbound packet a filter referencing non-packet data; and

a look-ahead function responsive to said filter for identifying a target application for said inbound packet.

44. A program storage device readable by a machine, tangibly embodying a program of instructions executable by a machine to perform method steps for control and management of communication traffic, said method steps comprising:

expressing access rules as filters referencing system

kernel data;

for outbound processing, determining a source application;

for inbound packet processing, executing a look-ahead function to determine a target application; and

responsive to said source or target application, executing filter processing.

45. A program storage device readable by a machine, tangibly embodying a program of instructions executable by a machine to perform method steps for control and management of aspects of communication traffic within filtering, said method steps comprising:

receiving IP packet data into a TCP/IP protocol stack executing within a system kernel

executing filtering code within said system kernel with respect to non-IP packet data accessed within said system kernel outside of said TCP/IP protocol stack.

46. A program storage device readable by a machine, tangibly embodying a program of instructions executable by a machine to perform method steps for centralizing system-wide communication management and control within filter rules, said method steps comprising:

providing filter statements syntax for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values; and

said selector referencing data that does not exist in IP packets.

47. A program storage device readable by a machine, tangibly embodying a program of instructions executable by a machine to perform method steps for managing and controlling communication traffic by centralizing access rules in filters executing within and referencing data available in system kernels, said method steps comprising:

receiving said packet in the kernel of the operating system of said first node from an application or process at said first node;

processing said packet by determining a task ID;

responsive to said task ID, determining a corresponding work control block;

responsive to said work control block, determining a process or job identifier;

responsive to said process or job identifier, determining job or process attributes.

48. The program storage device of claim 47, said method steps further comprising for inbound packet processing from said second node to said first node:

initially operating said kernel at said first node to determine a target application for said packet at said first node.

49. A computer program product or computer program element for control and management of communication traffic according to the steps comprising:

expressing access rules as filters referencing system kernel data;

for outbound processing, determining a source application;

for inbound packet processing, executing a look-ahead function to determine a target application; and

responsive to said source or target application, executing filter processing.

50. A computer program product or computer program element for control and management of aspects of communication traffic within filtering according to steps comprising:

receiving IP packet data into a TCP/IP protocol stack executing within a system kernel

executing filtering code within said system kernel with respect to non-IP packet data accessed within said system kernel outside of said TCP/IP protocol stack.

51. A computer program product or computer program element for centralizing system-wide communication management and control within filter rules according to method steps comprising:

providing filter statements syntax for accepting parameters in the form of a selector, each selector specifying selector field, operator, and a set of values; and

said selector referencing data that does not exist in IP packets.

52. A computer program product or computer program element for managing and controlling communication traffic by centralizing access rules in filters executing within and referencing data available in system kernels according to method steps comprising:

receiving said packet in the kernel of the operating system of said first node from an application or process at said first node;

processing said packet by determining a task ID;

responsive to said task ID, determining a corresponding work control block;

responsive to said work control block, determining a process or job identifier;

responsive to said process or job identifier, determining job or process attributes.

53. The computer program product or element of claim 52, said method steps further comprising for inbound packet processing from said second node to said first node:

initially operating said kernel at said first node to determine a target application for said packet at said first node.